

Exploiting the Semantic Similarity of Contextual Situations for Pre-filtering Recommendation

Victor Codina¹, Francesco Ricci², and Luigi Ceccaroni^{3,1}

¹Department of Software, Universitat Politècnica de Catalunya-BarcelonaTech (UPC)
Jordi Girona 1-3, K2M Building, Office 201, 08034 Barcelona, Spain
{vcodina, luigi}@lsi.upc.edu

²Faculty of Computer Science, Free University of Bozen-Bolzano,
Piazza Domenicani 3, 39100 Bolzano, Italy
fricci@unibz.it

³Barcelona Digital Technology Centre
Roc Boronat 117, 08018 Barcelona, Spain
lceccaroni@bdigital.org

Abstract. Context-aware recommender systems aim at outperforming traditional context-free recommenders by exploiting information about the context under which the users' ratings are acquired. In this paper we present a novel contextual pre-filtering approach that takes advantage of the semantic similarities between contextual situations. For assessing context similarity we rely only on the available users' ratings and we deem as similar two contextual situations that are influencing in a similar way the user's rating behavior. We present an extensive comparative evaluation of the proposed approach using several contextually-tagged ratings data sets. We show that it outperforms state-of-the-art context-aware recommendation techniques.

Keywords: Recommenders, Implicit Semantics, Collaborative Filtering, Matrix Factorization.

1 Introduction

Context-Aware Recommender Systems (CARSS) differ from traditional recommenders because when they estimate the rating of a target user u for an item i they do not only use a data set of ratings (of users for items), but they also exploit both the knowledge of the contextual conditions under which the ratings were acquired and the contextual situation of the target user asking for a recommendation [2]. We use the term *contextual factor* referring to a specific *type* of contextual information (e.g. weather), and *contextual condition* referring to a specific value for a contextual factor (e.g. sunny). The term *contextual situation* refers to a specific set of these contextual conditions that describe the context in which the user consumed and rated the item.

In the last years several CARS techniques have been proposed. We will illustrate some of them in the next section. A common limitation of these solutions resides in their requirement for a large data set of contextually tagged ratings, i.e., ratings for

items provided in all the various contextual situations that may be encountered by a user while experiencing an item.

A solution to this problem is, when making recommendations in a particular situation, to consider as relevant not only the ratings provided by the users in that situation but also to “reuse” ratings provided in similar situations. For instance, if we want to predict the rating for a place of interest, e.g., the South Tyrol Museum of Archaeology (in Bolzano, Italy), and the target contextual situation includes a condition such as, “group composition is two adults and two children”, ratings acquired when the group composition was “two adults and three children” may still be used to generate an accurate predictive model in the target situation. But, what about ratings acquired while the weather was sunny? A sunny day and a family do not seem to be comparable contextual conditions.

Actually, we have developed an approach to estimate the similarity of apparently unrelated contextual conditions, by focusing on the effect of the contextual conditions on the ratings provided by the users in those cases. The similarity among contextual conditions is estimated by identifying the “meaning” of a condition by means of its implicit semantics: that is, the meaning of a concept (here, a contextual condition) is captured by the usage of the concept. Hence, contextual conditions are similar if they co-occur and produce a similar effect on the user’s rating behavior. For example, in the recommendation of places of interest, weather conditions such as *cold* and *rainy* may have a positive effect on users’ ratings for indoor places like museums, and a negative effect for outdoor places like castles. In this scenario, if *cold* and *rainy* are similarly influencing user’s ratings on the same type of items, we consider them as semantically similar.

In a previous work it was shown that the accuracy of traditional recommenders can be improved by exploiting the implicit semantics of item attributes. For instance, Mobasher et al. [17] use Latent Semantic Analysis (LSA) [12] over the item-attribute matrix to calculate the semantic similarity between items, which then are used to boost an item-based Collaborative Filtering (CF) approach. Furthermore, in a more recent work [10], we show that the implicit semantic similarities between attributes captured in terms of how similarly users are interested in them (according to the user profiles previously learnt from user ratings), can be used to enhance a Content-Based (CB) prediction method [3]. Similarly, in this paper we show that the implicit semantics of contextual conditions can be exploited to implement an effective contextual pre-filtering approach that provides a better rating prediction accuracy compared with a standard two-dimensional (2D) Matrix Factorization (MF) method [15] and other state-of-the-art context-aware recommendation techniques.

The remainder of this paper is organized as follows. Section 2 positions our work with respect to the state of the art. Section 3 presents the method for acquiring the implicit semantics of contextual conditions. Section 4 describes the novel pre-filtering approach. Section 5 and 6 present the experimental evaluation of the proposed approach. Finally, section 7 draws the main conclusions and describes the future work.

2 Related Work

In [2], three different paradigms for incorporating contextual information into the recommendation process are presented (see Fig.1): (1) contextual pre-filtering, where context is used for selecting the relevant set of rating data before computing predictions; (2) contextual post-filtering, where context is used to adjust predictions generated by a context-free 2D prediction model; and (3) contextual modeling, in which contextual information is directly incorporated in the prediction model, usually by generalizing the 2D prediction model to an N-dimensional one.

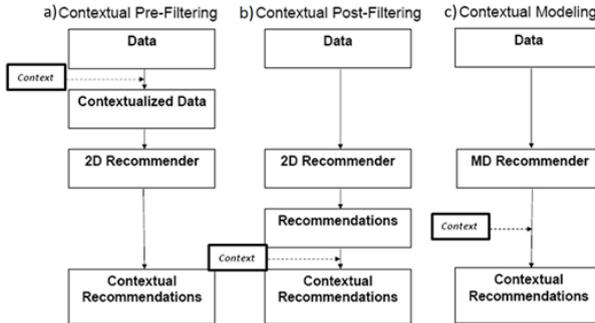


Fig. 1. Paradigms for incorporating context into the recommendation process

Among these paradigms, pre-filtering is especially appealing because it has a straightforward justification: when context matters, use in the recommendation process only the data acquired in the same contextual situation of the target user, because only this data is relevant for predicting user preferences. However, pre-filtering is not always the best option [18]. Its main limitation comes from the difficulty to obtain enough ratings in all the possible contextual situations in order to build a robust and largely applicable context-aware prediction model.

Adomavicius et al. [1] proposed a solution to this problem (also known as the *generalized* pre-filtering) that is based on the hierarchical structure of contextual information. In this approach more general contextual segments are used when there are not enough ratings in the specific target contextual situation. For instance, the *week-end* segment may be used when the target context is *Saturday* and not enough ratings have been acquired in that condition. As in the ontology-based approaches that exploit the explicit semantic relationships between item attributes to enhance the accuracy of CB recommendation [9][20], generalized pre-filtering strongly depends on the quality of the available domain-specific ontology, which is usually limited and only defines explicit relations between conditions of the same contextual factor. Moreover, it may produce a too coarse aggregation of contextual situations if the chosen contextual segments are too broad.

Another solution was proposed by Baltrunas and Ricci [8]. It is based on the idea of “item splitting” which consists of identifying in a selective way the relevant contextual conditions for the rating prediction of each single item. Hence, here ratings’

filtering is selectively carried out item by item. Similarly, Baltrunas and Amatriain [4] proposed the idea of “user splitting” which splits the user profile into several sub-profiles, each representing the given user in a particular context. More recently, Zheng et al. [21] introduced a feature selection method that searches for the most relevant contextual factors, which are then applied to specific components of a user-based CF approach [14].

However, all previously mentioned solutions do not fully exploit the potential similarities between contextual situations. For instance, a *mild* temperature in a *cloudy* day may have, on the user evaluation of a place of interest, the same effect of a *cooler* but *sunny* day. In this paper we present a pre-filtering approach that is based on the intuition that ratings acquired in contextual situations which have on the user’s rating behavior an *effect* similar to that produced by the target context may be used to improve the rating prediction accuracy. Similarly to user-based CF, which predicts the target user’s ratings with a local regression on similar users’ ratings, we predict here context-dependent ratings by regressing predictions computed in contextual situations similar to the target one.

3 Acquiring Implicit Semantics from Rating Data

In order to calculate the semantic similarities between contextual conditions we use the Vector Space Model (VSM), with an approach similar to that used in Information Retrieval to calculate similarities between two terms from a given corpus of documents [16]. It relies on the distributional hypothesis: terms repeatedly co-occurring in the same linguistic context (e.g. document, paragraph) tend to have similar meaning. Calculating semantic similarities between terms is divided in three steps: (1) term-document matrix (TDM) generation, where each entry stores the frequency, or a frequency-based weight (e.g. entropy), of a term in a document; (2) TDM transformation, which usually consists of reducing the matrix dimensionality, and (3) vector-matching calculation, where term vectors are compared using a vector-based similarity measure. A well-known example of this approach is LSA, where Singular Value Decomposition (SVD) is applied to the TDM for decomposing it, the term weights are computed with the entropy weighting function, and the cosine similarity is used to calculate vector similarity.

We have adapted these techniques to learn the semantic similarities of contextual conditions from contextually tagged rating data by considering two *contextual conditions as similar if they co-occur and influence the users’ ratings in a similar way*. To accomplish this task, we have adapted the step of TDM generation proposing two different methods for measuring the influence of contextual conditions on the users’ rating behavior: (1) an *item-based* method, in which a condition-item matrix is generated, terms (rows) correspond to contextual conditions, and documents (columns) to domain items, (2) a *user-based* method, in which a condition-user matrix is generated, and columns correspond to system’s users. In both methods each element of the matrix stores a weight representing the general influence (positive, negative or neutral) of the condition; the methods differ in how this weight is computed. In the *item-based*

method we measure the influence of conditions in an item-centered perspective, and the weight for a specific contextual condition c and item i (w_{ci}) is calculated as the average of the normalized ratings r_{uic} of the item i acquired in the contextual condition c . Denoting with R_{ic} this set of ratings, we have:

$$w_{ci} = \sum_{r_{uic} \in R_{ic}} (r_{uic} - b_{ui}) \frac{1}{|R_{ic}|} \quad (1)$$

where $b_{ui} = \mu + b_u + b_i$ is the baseline predictor [15] and is used to normalize the item ratings by subtracting the component of the rating prediction that does not take into account the contextual condition. Here μ is the overall rating average, b_u is the bias associated to the user u , and b_i the bias associated to the item i .

In the *user-based* method, the context influence is measured in a user-centered perspective, calculating the weight for a specific condition and user (w_{cu}) as the average of the normalized ratings r_{uic} given by the user u in that specific contextual condition c . If R_{uc} denotes this set of ratings, then we have:

$$w_{cu} = \sum_{r_{uic} \in R_{uc}} (r_{uic} - b_{ui}) \frac{1}{|R_{uc}|} \quad (2)$$

Once we have generated the condition-item or condition-user matrix M , we reduce its dimensionality using SVD. In a previous experimentation [11], we obtained the best results by setting the number of latent factors equal to the number of contextual conditions. Moreover, in the exceptional situation that the number of conditions is greater than the number of items or users, not transforming M yielded better results. We must note that in the considered data sets the number of conditions is small, and using an even smaller number of factors causes an information loss when computing the similarities between conditions.

Finally, we use the cosine measure to calculate the similarity between the semantic vectors of two contextual conditions. Table 1 shows, as an example, the top-5 similar conditions (and their corresponding similarity values) to the condition *cold* calculated by each method using the ratings of the tourism data set used for the experimentation (see Section 5.1 for more details on this data set).

Table 1. Top-5 similar conditions to the condition *cold* (in tourism data set)

Item-based	User-based
cloudy (0.84)	lazy (0.57)
working-day (0.81)	social-event (0.56)
rainy (0.79)	cloudy (0.49)
winter (0.75)	health-care (0.48)
night-time (0.71)	sad (0.47)

4 Exploiting the Implicit Semantics During Pre-filtering

The previously described semantic similarities between contextual conditions can be exploited to implement a novel pre-filtering approach that overcomes the stated limitation of the contextual pre-filtering paradigm, that is, the difficulty of producing

predictions when rating data are missing in the target contextual situation. When the system is requested to compute recommendations in a target context we first identify the ratings that can be used to build a predictive model in that situation and then we compute a standard MF model based on the collected ratings.

Fig. 2 shows the algorithm used to perform the first step of the process described above, i.e., collecting the ratings relevant to a target situation. Here a contextual situation is modeled as a vector of k contextual factors, that is, $c = [c_1, \dots, c_k]$, where $c_j = 0, 1, \dots, z_j$ and $c_j = 0$ means that the j -th factor value (condition) is unknown, while the other indexes refer to possible values of the factor. Given a training set of rating data provided by users for items in various contextual situations (Y), and given a target contextual situation (c^*), all the available ratings are sequentially scanned and all the ratings acquired in a contextual situation equal or similar enough to the target one are added to the training set (X). A convenient threshold (t) defines the minimum similarity value for a contextual situation to be considered as usable in a target context.

```

Input
  Y: set of contextually tagged ratings
  c*: target context consisting of k contextual conditions
  t: similarity threshold
Output
  X: set of ratings relevant to the target context
for each (u,i) with ratings in Y do
  if ( $\exists r_{uic} : c == c^*$ ) then
    add(X,  $r_{uic}$ );
  else
    # if  $r_{uic^*}$  is missing in Y
    for each  $r_{uic}$  in Y do
      if ( $Sim(c, c^*) \geq t$ ) then
        ratingSum +=  $r_{uic}$ ;
        nSimilar++;
      end if
    end for
    if (nSimilar > 0) then
      add(X, ratingSum / nSimilar); # Rating average
    else if ( $\exists r_{ui}$  in Y) then
      add(X,  $r_{ui}$ ); # Rating without specific context
    end if
  end if
end for

```

Fig. 2. Semantically-enhanced pre-filtering algorithm for rating collection

Note that when we identify ratings acquired in similar contextual situations, it is possible to identify, for a given user and item, more than one rating given by that user to that item in contextual situations similar to the target one. In this case, before adding them to X , we average these ratings, in order to generate a unique rating for a given user, item and target contextual situation triple. Finally, if we do not find any rating in conditions similar to the target one, we add to X the rating (if available) given by the user to that item not tagged with a specific context. In order to calculate the similarity between two contextual situations defined by multiple conditions, different *strategies* based on the *pairwise comparison* of the similarities of elementary contextual conditions can be used. We have evaluated two of them: (1)

best-pairs (summarized in Fig. 3), in which, for each contextual condition in the candidate rating situation, the best matching condition in the target situation is selected and then the similarities of these best pairs of conditions are averaged; and (2) *all-pairs* (defined in Fig. 4), where the similarities of all the pairs of conditions in the two contextual situations (target and candidate rating) are averaged. Note that the best-pairs similarity is asymmetric, that is $Sim(c,c^*) \neq Sim(c^*,c)$, and the all-pairs similarity is symmetric.

```

Input
  c: candidate context
  c': target context
Output
  s: similarity value (real)
for each (cj in c AND cj!=0) do
  ci' = getMostSimilarCondition(c',cj);
  s = s + getConditionSimilarity(cj,ci');
  nMatchings++;
end for
return s / nMatchings; # Average value

```

Fig. 3. Best-pairs pairwise strategy method

```

Input
  c: candidate context
  c': target context
Output
  s: similarity value (real)
for each (cj in c AND cj!=0) do
  for each (ci' in c' AND ci'!=0) do
  s = s + getConditionSimilarity(cj,ci');
  nMatchings++;
  end for
end for
return s / nMatchings;

```

Fig. 3. All-pairs pairwise strategy method

5 Experimental Setup

5.1 The Data Sets

We have considered five real-world contextually tagged data sets. Table 2 shows their main characteristics. The Music and Tourism data sets contain multiple ratings for the same item and user, in contextual situations described by only one condition. The music data set contains eight contextual factors such as *driving style*, *mood* and *landscape*; the tourism one contains 14 factors such as *weather*, *companion* and *travel goal*. Each factor has multiple contextual conditions. For instance, *active*, *passive*, *happy* and *sad* are possible conditions of the *mood* factor. More details about these data sets can be found in [5] and [6].

“MovieAdom” is derived from the data set used by Adomavicius et al. [1]. The ratings were collected in a survey of college students who also provided information about the context of the movie-watching experience. In our experiments we used four

contextual factors: *companion*, *day of the week*, *movie venue*, and if it was on the *opening weekend*. Differently from the previous data sets, in this case contextual situations are actually described by the conjunction of several contextual conditions.

The LibraryThing and MovieLens are data sets about books and movies, respectively, and ratings are augmented with user tags. They were extracted from their respective websites¹. Here the contextual situation is given by the tags assigned by the users to the rated items: they provide a contextual clue of why the item is important for the user. Given the high number of distinct tags available and the inherent noise of tag information, we only used as contextual factors the 100 most popular tags, i.e., those with the largest number of users that used the tag at least once.

Table 2. Data sets' statistics

Data set	Users	Items	Factors	Ratings	Scale
Music	43	139	8	4013	1-5
Tourism	25	20	14	1679	1-5
MovieAdom	84	192	4	1464	1-13
MovieLens	849	2207	100	6784	1-10
LibraryThing	7112	37K	100	626K	1-10

5.2 The Evaluation Protocol

As suggested by Shani and Gunawardana [19], we measured the rating prediction accuracy of the considered models by conducting a per-user evaluation schema where, for each user, five ratings were randomly chosen as test set to compute the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE).

In order to avoid the new-user cold-start problem when evaluating the pre-filtering approaches, we selected as testing ratings those provided in contextual situations in which the user rated at least one additional item. As a result of this data partition, only some users could be finally considered for testing: 18 (42% of the users) in the music data set, 8 (32%) in the tourism one, 25 (30%) in MovieAdom, 71 (8%) in MovieLens and 2219 (31%) in LibraryThing. The remaining users and ratings were used to train the prediction models and to acquire the implicit semantics of contextual conditions. We also tried other variations for splitting the data but obtained quite similar results.

All the reported results are averages of per-user evaluations, and the statistical significance of the differences between the proposed methods and the baseline algorithms have been calculated by means of the paired Wilcoxon sign rank test.

5.3 Prediction Models

We have compared the performance of the proposed semantically enhanced, pre-filtering approach to four state-of-the-art context-aware CF models: a MF-based *exact* pre-filtering method, which generates rating predictions for a target contextual

¹ See [www.librarything.com] and [www.movielens.org], accessed January 6, 2013.

situation by considering only the ratings acquired in that situation; *CAMF-CC* and *CAMF-CI*, two model-based context-aware MF approaches proposed in [7]; and the N-dimensional Tensor Factorization (TF) model proposed in [13], which also follows the contextual modeling paradigm.

In particular, the pre-filtering approaches presented here generate rating predictions in a target contextual situation using the bias-based MF proposed in [15], but employing different methods for filtering the training ratings as explained above. In bias-based MF the rating estimation for the user u and item i is:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u \quad (3)$$

where q_i is the factor vector of item i , and p_u is the factor vector of user u . To learn the model parameters we minimized the *regularized error* using stochastic gradient descent, which has been proved to be an effective approach [15]. We minimized the *absolute loss* function when using the MAE and the *squared loss* function for RMSE. When we applied bias-based MF, we always used a hyper-parameters' configuration [11] with which the models converged within 200 iterations in the music, tourism and MovieAdom data sets, 50 in MovieLens, and 25 in the LibraryThing data set. The pre-filtering models that we have compared are the following:

- *Exact-Prefiltering* is the *exact* pre-filtering method that trains the baseline MF in a target context using only the ratings whose context exactly matches the target one.
- *Semantic-Prefiltering-gt* and *Semantic-Prefiltering-dt* are two variants of the proposed pre-filtering approach that exploit the semantic similarities of contextual conditions (see Fig. 2 for more details). *gt* refers to the variant that uses a global similarity threshold for all targets' contextual situations, and *dt* corresponds to a variant that employs a different threshold per target context. In both variants, the optimal threshold values are learned in the training set by using the same per-user evaluation schema, but choosing three ratings for validation instead of five.
- *Semantic-Prefiltering-bp* and *Semantic-Prefiltering-ap* are the two further variants of *Semantic-Prefiltering-gt*: *bp* uses the *best-pairs* strategy (see Fig. 3) while *ap* uses the *all-pairs* strategy (see Fig. 4) for computing the similarity between contextual situations defined by multiple conditions. We chose *gt variant* for comparing the pairwise strategies to reduce the computational cost of the experiments.

The other rating prediction models included in the experimentation are:

- *User-Item-avg*, the non-personalized and not context-aware baseline prediction model defined by $\hat{r}_{ui} = \mu + b_u + b_i$;
- *MF*, the baseline 2D bias-based MF model whose estimation function is defined by Equation 3;
- *CAMF-CC* and *CAMF-CI*, the context-aware MF models proposed by Baltrunas et al. [7];
- *TF*, the N-dimensional Tensor Factorization model proposed by Karatzoglou et al. [13], which we used only for the MovieAdom data set due to the high computational cost of this approach when the number of contextual factors is greater than five (MovieAdom contains four factors).

6 Results and Discussion

Tables 3 and 4 show the main results of the prediction models evaluated. We note that in these experiments the semantically-enhanced variants use the *item-based* method for acquiring the context similarities, which produces better results compared to the *user-based* method.

The *Semantic-Prefiltering-gt* and *Semantic-Prefiltering-dt* algorithms can be used only when the contextual situation is defined by a single contextual condition, hence, only in the music and tourism data sets (see Table 3). The results show that *Semantic-Prefiltering-gt* is better than *Semantic-Prefiltering-dt*. *Semantic-Prefiltering-gt* reduces the MAE of MF by 16% (music data set) and 30% (tourism). We believe that the difference in the performance of these two variants can be related to the fact that learning optimal similarity thresholds for each target situation (a single condition in these data sets) can over-fit the small validation data set and therefore perform poorly on the testing data. *Semantic-Prefiltering-gt* also performs better than the other context-aware models.

Exact-Prefiltering reduces the MAE of MF by 12% and by 24% in the music and tourism data sets, respectively, while model-based approaches (*CAMF-CC*; *CAMF-CI*) only reduce the error of MF in the tourism data set. These results differ from the ones reported in [7] for the same data sets; we believe that these differences are due to the different evaluation protocols employed. An extended comparison of the models and the significance test results can be found in [11].

Table 3. Performance of evaluated models in the music and tourism data sets. Results in **bold** are statistically significant better (95% of confidence level) than the context-free models.

Prediction Model	Music		Tourism	
	RMSE	MAE	RMSE	MAE
<i>User-Item-avg</i>	1.186	.978	1.193	1.030
<i>MF</i>	.825	.573	1.124	.959
<i>CAMF-CC</i>	.858	.599	.966	.775
<i>CAMF-CI</i>	.912	.601	1.037	.779
<i>Exact-Prefiltering</i>	.769	.503	.986	.729
<i>Semantic-Prefiltering-gt</i>	.755	.480	.944	.673
<i>Semantic-Prefiltering-dt</i>	.791	.483	.988	.715

When considering the *Semantic-Prefiltering-bp* and *Semantic-Prefiltering-ap* variants we focus on their performance on the data sets where the rating context is defined by multiple conditions: the MovieAdom, the MovieLens and the LibraryThing data sets (see Table 4). The experimental results show that *Semantic-Prefiltering-ap* (*all-pairs* strategy) is more effective than *Semantic-Prefiltering-bp* (*best-pairs* strategy). We note that, in the MovieLens data set, the differences between the two variants are not significant. This is due to the fact that in this data set only a small percentage of the target contexts are defined by multiple conditions, hence it is not an unexpected result.

Table 4. Performance results in the data sets with multiple conditions per context. Results in **bold** are statistically significant better (95% of confidence level) than the context-free models. Results underlined are also significantly better than the other context-aware models.

Prediction Model	MovieAdom		MovieLens		LibraryThing	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
<i>User-Item-avg</i>	2.459	1.990	1.583	1.332	1.432	1.213
<i>MF</i>	2.092	1.719	1.584	1.342	1.415	1.208
<i>CAMF-CI</i>	2.192	1.823	1.612	1.365	1.466	1.230
<i>TF</i>	2.292	1.928	-	-	-	-
<i>Exact-Prefiltering</i>	2.040	1.714	1.426	1.191	1.532	1.287
<i>Semantic-Prefiltering-bp</i>	2.210	1.825	1.428	1.176	1.415	<u>1.188</u>
<i>Semantic-Prefiltering-ap</i>	2.013	1.713	1.427	1.175	<u>1.405</u>	<u>1.179</u>

In this scenario only *Semantic-Prefiltering-ap* always improves the performance of *MF*: the most significant improvement is in the MovieLens data set, where we have obtained a 12% reduction of MAE. In contrast, *CAMF-CI* and *TF* do not improve the performance of *MF*. We believe these poor results may be due to the low relevance of contextual factors used in the data sets (above all the tag-based ones).

7 Conclusions and Future Work

We have presented a novel recommendation approach that overcomes the major limitation of contextual pre-filtering paradigm, i.e., using only user data acquired in the target context when the system is requested to make a recommendation in that target situation. Our solution relies on the identification and usage of ratings acquired in contextual situations that are semantically similar to the target one. We have introduced a novel notion of similarity among contextual situations that is based on how similarly two different situations are influencing the users' ratings. We have presented several variants of the proposed pre-filtering approach and experimentally compared their performance. The experimental results on several contextually tagged data sets show that our approach outperforms other state-of-the-art context-aware Matrix Factorization (MF) models.

As future work we plan to extend the performance comparison by investigating new variants of the pre-filtering approach presented here and by using larger real-world, contextually-tagged data sets. In addition, we will improve our experimentation on the tag-based data sets by filtering irrelevant tags, on the base of a statistical significance test such as the Pearson's chi-squared.

Acknowledgments. The research described in this paper is partly supported by the SuperHub and the Citeclops European projects (FP7-ICT-2011-7, FP7-ENV-308469), and the Universitat Politècnica de Catalunya (UPC) under an FPI-UPC grant. The opinions expressed in this paper are those of the authors and are not necessarily those of SuperHub or Citeclops projects' partners.

References

1. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems* 23(1), 103–145 (2005)
2. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.-B. (eds.) *Recommender Systems Handbook*, pp. 217–250. Springer (2011)
3. Ahn, J., Brusilovsky, P., Grady, J., He, D., Syn, S.: Open user profiles for adaptive news systems: help or harm? In: *Proceedings of the 16th International Conference on World Wide Web*, pp. 11–20 (2007)
4. Baltrunas, L., Amatriain, X.: Towards time-dependant recommendation based on implicit feedback. In: *Workshop on Context-Aware Recommender Systems, CARS 2009* (2009)
5. Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., Lüke, K.-H., Schwaiger, R.: InCarMusic: Context-Aware Music Recommendations in a Car. In: Hummer, C., Setzer, T. (eds.) *EC-Web 2011. LNBIP*, vol. 85, pp. 89–100. Springer, Heidelberg (2011)
6. Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing* 16(5), 507–526 (2012)
7. Baltrunas, L., Ludwig, B., Ricci, F.: Matrix factorization techniques for context aware recommendation. In: *Proceedings of the 2011 ACM Conference on Recommender Systems, Chicago*, pp. 301–304 (2011)
8. Baltrunas, L., Ricci, F.: Context-dependent items generation in collaborative filtering. In: *Proceedings of the 2009 ACM Conference on Recommender Systems, New York*, pp. 245–249 (2009)
9. Cantador, I., Castells, P., Bellogín, A.: An Enhanced Semantic Layer for Hybrid Recommender Systems: Application to News Recommendation. *International Journal on Semantic Web and Information Systems* 7(1), 44–77 (2011)
10. Codina, V., Ceccaroni, L.: Semantically-Enhanced Recommenders. In: *Proceedings of the 15th International Conference of the Catalan Association for Artificial Intelligence*, pp. 69–78. IOS Press (2012)
11. Codina, V., Ricci, F., Ceccaroni, L.: Semantically-Enhanced Pre-filtering for Context-Aware Recommender Systems. In: *3rd ACM Workshop on Context-Awareness in Retrieval and Recommendation, CaRR 2013, Rome, Italy*, pp. 15–18 (2013)
12. Dumais, S.: LSA and information retrieval: Getting back to basics. In: Landauer, T.-K., McNamara, D.-S., Dennis, S., Kintsch, W. (eds.) *LSA: A Road to Meaning*, pp. 293–321. Lawrence Erlbaum (2006)
13. Karatzoglou, A., Amatriain, X., Baltrunas, L., Olivier, N.: Multiverse Recommendation: N-dimensional Tensor Factorization for Context-aware Collaborative Filtering. In: *Proceedings of the 2010 ACM Conference on Recommender Systems, Barcelona, Spain*, pp. 79–86 (2010)
14. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM* 40(3), 77–87 (1997)
15. Koren, Y., Bell, R.: Advances in Collaborative Filtering. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.-B. (eds.) *Recommender Systems Handbook*, pp. 145–186. Springer (2011)
16. Manning, C., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*. Cambridge University Press (2008)

17. Mobasher, B., Jin, X., Zhou, Y.: Semantically Enhanced Collaborative Filtering on the Web. In: Berendt, B., Hotho, A., Mladenič, D., van Someren, M., Spiliopoulou, M., Stumme, G. (eds.) EWMF 2003. LNCS (LNAI), vol. 3209, pp. 57–76. Springer, Heidelberg (2004)
18. Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., Pedone, A.: Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. In: Proceedings of the 2009 ACM Conference on Recommender Systems, New York, pp. 265–268 (2009)
19. Shani, G., Gunawardana, A.: Evaluating Recommendation Systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.-B. (eds.) Recommender Systems Handbook, pp. 257–297. Springer (2011)
20. Sieg, A., Mobasher, B., Burke, R.: Ontological User Profiles for Personalized Web Search. In: Proceedings of the AAAI 2007 Workshop on Intelligent Techniques for Web Personalization, pp. 84–91 (2007)
21. Zheng, Y., Burke, R., Mobasher, B.: Optimal feature selection for context-aware recommendation using differential relaxation. In: 4th ACM Workshop on Context-Aware Recommender Systems, CARS 2012 (2012)